

18/5/07

1

# **Evolving structure-function mappings in cognitive neuroscience using genetic programming**

**Fernand Gobet**

Centre for Cognition and Neuroimaging

School of Social Sciences and Law

Brunel University

**Amanda Parker**

School of Biology

Psychology, Brain & Behaviour

University of Newcastle

## **Send correspondence to**

Fernand Gobet

Centre for Cognition and Neuroimaging

School of Social Sciences and Law

Brunel University

Uxbridge

Middlesex, UB8 3PH

United Kingdom

Fax: (+44)(0)1895 237573

[fernand.gobet@brunel.ac.uk](mailto:fernand.gobet@brunel.ac.uk)

**Running head: Evolving structure-function mappings**

### **Abstract**

A challenging goal of psychology and neuroscience is to map cognitive functions onto neuroanatomical structures. This paper shows how computational methods based upon evolutionary algorithms can facilitate the search for satisfactory mappings by efficiently combining constraints from neuroanatomy and physiology (the structures) with constraints from behavioural experiments (the functions). This methodology involves creation of a database coding for known neuroanatomical and physiological constraints, for mental programs made of primitive cognitive functions, and for typical experiments with their behavioural results. The evolutionary algorithms evolve theories mapping structures to functions in order to optimize the fit with the actual data. These theories lead to new, empirically testable predictions. The role of the prefrontal cortex in humans is discussed as an example. This methodology can be applied to the study of structures or functions alone, and can also be used to study other complex systems.

### **Keywords**

Complex systems, evolutionary computation, prefrontal cortex, scientific discovery, structure-function mapping, theory formation

## Evolving structure-function mappings in cognitive neuroscience using genetic programming

A primary aim in science is to develop theories that summarize and unify a large body of experimental data. However, there is no overarching theory in psychology (or even in subfields of psychology, such as the study of memory, emotions or perception) which imposes order on the mass of data and makes it possible to derive quantitative predictions, in the way, for example, quantum mechanics can be used to organize empirical data in chemistry. To compound the difficulty, there are currently around 1,500 journals devoted to scientific psychology. A substantial proportion of these journals publish mainly experimental results. Psychology is not exceptional: in sciences from astrophysics to meteorology to biology, technological progress has enabled the rapid collection of huge amounts of data. How can scientists keep track of this exponentially increasing amount of information, in spite of their bounded rationality?

While progress in database management of scientific results is notable, there remains the question of how this new information can foster scientific *understanding*, as opposed to simple *accumulation* of knowledge. A particularly interesting approach is to develop theories implemented as computer programs which account for, and therefore summarize, empirical data.

How scientific theories are developed has been the focus of a number of studies in psychology, philosophy, history, and, more recently, artificial

intelligence (e.g., Hanson, 1958; Kuhn, 1977; Langley, Simon, Bradshaw, & Zytkow, 1987; Gholson, Shadish, Neimeyer, & Houts, 1989). In psychology, researchers have proposed a variety of explanations, including intuition (Hadamard, 1945), heuristic search (Simon, 1977), as well as random variation and Darwinian selection (Simonton, 1999). If the process of scientific discovery can be described precisely in information-processing terms, as argued forcefully by Langley et al. (1987), it should be possible to automate it—thus alleviating researchers' task of dealing with an exponentially increasing quantity of information.

The goal of this paper is to help psychologists and other scientists develop powerful theories explaining complex data. This is done by offering a methodology which, combined with other informatics techniques (Kanehisa, 2000; Koslow & Huerta, 1997), could help solve the problem of information overload in science. This methodology consists of powerful search methods (evolutionary computation) aimed at automatically developing theories. As an illustrative example, we have chosen the question of the localization of cognitive functions in the brain, and, more specifically, of the role of the prefrontal cortex—a difficult and topical research question. Given the readership of this journal, we are not so much interested in discussing the technical details underlying the computational implementation of our approach as in pointing the presence of potentially powerful methods to psychologists and philosophers.

### **The role of the prefrontal cortex in humans**

An important goal in biological psychology and neuroscience is to map cognitive processes onto neuroanatomical structures. Currently, a massive amount of data is being collected using such approaches as fMRI and lesion studies with non-human primates. However, neuroscientists have been increasingly worried about their limits to assimilate information from empirical research (Purpura, 1997). Research into the prefrontal cortex offers a striking example of this situation.

In line with research on monkeys (Parker & Gaffan, 1998), this area of the cortex is assumed to be involved in working memory, planning, decision making, the acquisition of behavioural strategies, emotions, and motivation (Roberts, Robbins, & Weiskrantz, 1998). However, progress in mapping structures to functions in the prefrontal cortex lags behind advances in those brain areas underpinning perceptual processing. Indeed, no current theory is able to account for even a small subset of the data—posing a sufficient challenge to test the validity of our methodology.

### **The problem of mapping brain structures to functions**

Structure-to-function mapping in the prefrontal cortex is difficult for three reasons:

1. Our understanding of prefrontal cortical anatomy (both functional and structural), particularly in humans, is less developed than for other parts of the brain (e.g., visual cortex).
2. Many different *mental programs* (i.e., organized sequences of serial and parallel cognitive processes) may instantiate the same pattern of behavioural data, making search through the space of possible mental programs difficult.
3. The mapping between the two domains, a difficult task in itself, is made more difficult by the uncertainties in our knowledge of human neuroanatomy and the processes involved in mental programs.

We therefore face a typical problem of optimizing search through multiple spaces. Hitherto, most research has addressed this question using informal theories, a weak method given the size of the search space. Moreover, modelling has not been applied to solving the mapping problem itself, but instead to provide functional mechanisms for simulating empirical data (Burgess & Shallice, 1996; Cohen, Braver & O'Reilly, 1998; Dehaene & Changeux, 1995). We propose that the mapping problem can be tackled by using the powerful search methods offered by evolutionary computation.

### **Computational scientific discovery**

Scientific research can be described as heuristic search in combinatorial spaces (Langley et al., 1987; Simon, 1977). 'Combinatorial' means that, at each

choice point, many decisions are possible. This combinatorial explosion implies that the search space drastically outgrows human capacities to explore it. To circumvent these limits, research in artificial intelligence has developed efficient search techniques in order to devise new laws, theories, and concepts. These techniques, used either autonomously or semi-autonomously, have been successfully applied in sciences such as chemistry, biology, mathematics (e.g., Bollobas & Riordan, 1998; Valdés-Pérez, 1999; Xu, Mural, Einstein, Shah, & Eberbacher, 1996).

One class of computational search techniques is evolutionary computation, which includes genetic algorithms (Goldberg, 1989; Holland, 1992; Mitchell, 1996) and genetic programming (Koza, 1992; 1994). Inspired by natural selection, evolutionary computation enacts a search for solutions to the problem of survival. It evolves large populations of genotypes (possible solutions) with the constraint that the fittest (best) tend to survive and reproduce. Artificial genotypes encode sets of parameters with genetic algorithms, or entire programs with genetic programming. What constitutes a ‘solution’ is governed by a fitness function determined by the specific problem at hand. For example, if the problem is to optimize the fit of a mathematical function to a set of empirical data, the amount of variance accounted for may be used as the fitness measure. The mathematical foundation of evolutionary computation is well established (Holland, 1992).

While computationally simple, these algorithms are robust and powerful. Being less sensitive to local minima and to initial conditions than other hill-climbing methods (Koza, 1992), they can explore huge search spaces efficiently and in parallel, even when the information is noisy and subject to uncertainty. These algorithms have been used extensively in science and engineering, for example in function optimisation, pattern recognition, functional genomics, and the analysis of noisy data (Kanehisa, 2000; Goldberg, 1989; Mitchell, 1996; Koza, 1992, 1994). Recent neuroscientific applications include diagnostic discovery (Kentala, Laurikkala, Pyykko, & Juhola, 1999), neuromuscular disorders assessment (Pattichis & Schizas, 1996), and interpretation of magnetic-resonance brain images (Sonka, Tadikonda, & Collins, 1996).

### **Genetic-programming algorithms**

We propose to use a variant of evolutionary computation called genetic programming (Koza, 1992, 1994). Instead of acting on digital chromosomes, as do genetic algorithms, genetic programming evolves entire computer programs. This removes some of the limitations of classical genetic algorithms (mainly, necessity to use fixed-length chromosomes, difficulty in representing hierarchical structures; and lack of dynamic variability; cf. Koza, 1992). The hierarchical structure of these programs is also convenient for simulating human mental programs.



---

Insert Figure 1 about here

---

Programs are represented as trees, a common data structure in computer science, to allow ease of manipulation and evolution. For example, the function  $(A + (B * 2))$  can be represented as a tree (Figure 1a). ‘A’, ‘B’, and ‘2’ are called *terminals* and ‘+’ and ‘\*’ are called *operators*. With simulated mental programs, operators consist of cognitive operations, and terminals of possible inputs. Figure 1b encodes an elementary mental program that compares two perceptual inputs, and then outputs the outcome of the comparison.

Genetic programming involves four steps:

1. *Choose a set of admissible operators and terminals.* In the case of mental programs, these are cognitive operations and possible inputs, respectively.
2. *Generate an initial population of trees.* These trees use the set of admissible operators and terminals, and can be of different sizes and shapes. They may be generated either fully randomly, or as random variations of plausible programs, which can be derived from the literature. Syntactically incorrect trees are deleted.

3. *Calculate the fitness of each program.* The fitness is computed based on the performance of each program on a set of problems ('fitness cases'), where both the input and the output are known. With mental programs, the fitness cases consist of behavioural data of humans performing prefrontal cortex tasks, where the fit is the amount of variance accounted for in the empirical data by the program.

4. *Apply selection, crossover, and mutation.* Selection involves keeping a predetermined number of programs in the population, selected probabilistically in proportion to fitness. Crossover refers to producing two offspring from a random point in each of the two parents and swapping the resultant subtrees (Figure 2). Mutation consists of replacing a subtree below a random point by a randomly created subtree. Typically, mutation has a much lower probability than crossover (Mitchell, 1996).

---

Insert Figure 2 about here

---

Genetic programming offers an efficient way of using feedback about a program's fitness within a generation to breed the next generation. While the basic representation uses trees, with each node having a single parent, several techniques exist to implement cyclic or recursive function calls, such as 'automatically defined functions' (Koza, 1994; Angeline & Pollack, 1992).

### **Searching through the space of mapping theories**

Complex systems, both natural and artificial, are more likely to evolve from simple systems if they are organized as modules and hierarchies—i.e., if they contain stable intermediate forms (e.g., Simon's, 1996, concept of near-decomposability). Such hierarchical organisations have often been proposed for brain structures (Churchland & Sejnowski, 1992), cognitive processes (Kosslyn & Koenig, 1992), and knowledge representations (Gobet, 2001; Gobet & Simon, 2000; Newell, 1990). While the assumption of brain modularity has been disputed (Elman et al., 1996; Uttal 2001), it is explicitly accepted by many researchers in neuroscience (Churchland & Sejnowski, 1992; Shallice, 1992) and implicitly by most researchers using neuro-imaging techniques.

We assume that the interactions between elements are high within a subsystem of the hierarchy, but weak between subsystems (Simon, 1996). This assumption is important in that it simplifies the search space of the structure-to-function mappings, because subsystems can be considered as units during simulated evolution. This also enables the direct use of genetic-programming subtree-encapsulation methods (Koza, 1994).

---

Insert Figure 3 about here

---

Figure 3 depicts the abstract relations formulated here. Four properties are important: (a) structures are organized hierarchically; (b) functions are organized hierarchically; (c) a one-to-one, one-to-many or a many-to-one mapping may link functions and structures (thus, the same elemental structure may implement multiple functions); and (d) in both hierarchies, some subtrees can be considered as modules.

A theory mapping structure to function (hereafter, mapping theory) comprises a 3-part system: (a) a hierarchy representing brain structures and their connections; (b) a hierarchy representing cognitive programs; and (c) the mapping between these two hierarchies. The same mapping theory is used across several tasks, providing an important constraint for selection.

Mapping theories are explicit, and therefore easily understood by humans. They can be manipulated and evolved as executable data structures. The key of our methodology is to *evolve* such mapping theories, using as a fitness function the extent to which they account for empirical data. While limiting the number of theories, thereby eliminating potentially successful theories, the mechanisms of evolutionary computation ensure that only viable theories are evolved. These mechanisms also lead to diverse, sometimes incompatible theories that all account for the data equally well. This is a strength of our method, as these competing theories and the new predictions that are derived from them can be tested empirically.

We can identify five main stages in our methodology:

1. *Codification of current knowledge of anatomical structures, their connections, and their hierarchical organisation.* This includes information about putative hierarchical organisation. The databases are coded using trees for their later input to evolutionary algorithms.

2. *Codification of current knowledge of cognitive processes implicated in prefrontal cortex tasks, and their hierarchical organisation.* Although complex, each cognitive process can be viewed as an operation or series of operations. Each operation has two parameters: list of input values, list of output values. Based on these operations, programs for classical prefrontal cortex tasks, such as discrimination reversal, delayed match to sample, or Wisconsin Card Sort, can be written. These programs use, as elementary operations, a limited set of processes gleaned from the literature (such as inhibition of visual information, maintenance of verbal information, matching of two visual pieces of information). The elementary processes are organized as subsystems, thus allowing a hierarchical organisation of the programs. There is no doubt that translating current theories into formal programs can result in a wide variety of different programs, possibly revealing inconsistencies across authors, or even within authors. These programs, used as the initial seed population, can be tested to ensure that they obtain performance similar to 'normal' participants, and thus are sufficient for the task.

3. *Preliminary mapping of the two types of knowledge.* These two knowledge bases, and their mapping, are combined into mapping theories, again guided by the current literature. Each operation has one or more cytoarchitectonic areas associated with it, as well as lists of input and output areas, and their respective values. Operations may function properly only if the necessary cortical area is intact, and if the input and output areas, as well as their connections, are intact (in all cases, impairment may be only partial). Information can also be encoded about the degree of confidence in the operation-area mapping.

4. *Construction of a database of tasks containing empirical results; these tasks are then used to compute the fitness function of the mapping theories.* To test the fit of the mapping theories to the empirical data requires the creation of a database of results from human studies using prefrontal cortex tasks, both with normal participants and brain-damaged patients. The database can also incorporate brain scan data. It is then possible to compute the fitness of a given theory by applying it to a set of tasks and to a set of neurological damages, and to compare the predictions of the theory with the empirical data, using standard measures of goodness of fit such as  $r^2$  (amount of variance accounted for).

5. *Use of genetic-programming techniques to optimize search between these two types of knowledge and their conjunction.* The final stage uses evolutionary algorithms to optimize search through the spaces of structure and function as well as their conjunction. Several approaches can be used to search for

new and better hypotheses for the mapping of structures to processes: search of the space of structures, processes, and mappings, respectively, while keeping the two other spaces constant; search of two spaces while keeping the third space constant; and, finally, search through the three spaces simultaneously. The role of the initial conditions and parameters of the system, such as mutation and crossover rate, can also be explored by systematically varying these conditions and parameters.

### **A simplified example**

As an illustration, consider the delayed match to sample (DMTS) task. In this task, a stimulus is first presented for a given amount of time, followed by a delay. Then, two stimuli are presented, and the task is to select which of these two stimuli matches the stimulus presented first. This task has been extensively used with humans and non-human animals. In this example, we used the data of Chao, Haxby and Martin (1999), who studied humans. We chose this paper because the task (a) is simple, (b) uses only known stimuli, which makes it unnecessary to model any learning process, and (c) is short enough so that forgetting does not need to be taken into consideration, at least as a first approximation. We focused on two conditions: in the first, pictures of animals were presented as stimuli, and, in the second, pictures of tools were used. We explain in some detail how our methodology can evolve (indeed, has evolved) functional theories with this task;

then, more briefly, we indicate how it could be used to evolve full mapping theories.

First, we programmed the task itself. We generated simplified versions of the stimuli used by Chao et al. (1999), by using pseudo-random sequences of digits. We then implemented the exact design in that paper, including the exact number of subjects, stimuli, and trials used.

Second, we selected a set of cognitive operations from the literature. These included operations for inputting a stimulus, storing information in short-term memory (STM), and matching two stimuli. Operations were not fully deterministic, as they had a .02 probability of failing to produce the correct output. Based on the literature (e.g., Cowan, 2001; Gobet & Clarkson, 2004), we assumed that visual STM was limited to four items.

Third, we defined a fitness function, using the deviation percentage from the means and standard deviations of the “animal” and “tool” conditions in Chao et al.’s (1999) study. The deviation percentage was defined as:

$$(|\text{Theory} - \text{Observed}| / \text{Observed}) * 100$$

where Observed refers to the human data, as provided by Chao et al., and Theory refers to the prediction of an evolved theory. As the experimental design is



exactly the same with the simulations as with the human data (see above), the observed and predicted means and standard deviations are directly comparable.

We also defined a hit measure. When an evolved theory obtained a value within 10% of the value observed with the human data, we arbitrarily counted this as a hit. For example, the condition “animal” yielded a value of 1.4% for the standard deviation; the accepted range is therefore (1.26% - 1.54%); thus, 1.50% would be counted as a hit, but 1.60% would be counted as a miss. In this example, the fitness function was the sum of relative deviations; other functions are possible, such as the sum of hits, or the amount of variance explained by linear regression ( $r^2$ ).

Fourth, we let a modified version of genetic programming evolve the theories, starting with a population made of random trees. The fitness of each evolved theory was computed by running through the two conditions of Chao et al.’s (1999) experiment. In spite of the fact that only four fitness cases were available, the program managed to evolve interesting theories. For example, after 14 generations, the program depicted in Figure 4 was produced. The top part of the Figure shows the (slightly edited) Lisp code of the theory evolved, and the bottom part a diagrammatic representation of this theory. In words, what this theory does is to store the first stimulus in STM, and then to store the second and third stimuli). Finally, the second and third elements in STM are compared. Because STM is implemented as a queue, new elements “push” older elements;

as a consequence, by the time the comparison takes place, stimulus #1 is in the third slot of STM, and stimulus #2 in the second slot. The theory does a good job at predicting means, but is wide of the mark with standard deviations (see Table 1). While rather modest, this theory shows that functional theories can be involved, at least in the conditions of these simulations.

---

Insert Figure 4 about here

---

---

Insert Table 1 about here

---

The methodology discussed in this paper involves one additional—and important—complication: to use theories that combine structure and function. Figure 5, which was constructed by the authors, and not generated automatically, shows such a possible mapping theory, extending the program shown in Figure 4. Note that the brain structure could be further specified; for example, the dorsolateral prefrontal cortex could be divided into Brodmann areas 9 and 46. Chao et al. (1999) do not present data about the prefrontal cortex, but brain-imaging or patient data available elsewhere in the literature could be used to compute a more sophisticated fitness function that would include the information about brain structures. For example, if the empirical data show that the

dorsolateral prefrontal cortex is not engaged in this task, contrary to what the program states, then this would count against (the fitness value of) the theory.

---

Insert Figure 5 about here

---

### **Role of constraints in reducing the search space**

Even when considering only the functional side of our approach, the number of possible mapping theories is theoretically infinite. In order to have practical chances to search this space successfully, it is necessary to use a number of heuristics that constrain search. For example, we have imposed a modular and hierarchical structure to the theories. We have also biased search by selecting the cognitive operations used, and not selecting others. In addition, we have chosen a given experiment, and not another. While the full application of our framework would incorporate more cognitive operations and more fitness cases, in addition to using full mapping theories, there is no doubt that (strong) assumptions will have to be used as constraints to make our method feasible—as is the case generally with optimization techniques. We refer the reader to the section “Searching through the space of mapping theories” for a discussion of some of

these, namely modularity and hierarchical organization of cognitive functions and brain structures.

### **Likely objections**

Our proposal is likely to raise a number of objections, and it is important to review some of them here.

*1. The search space is too large.* We agree that the search space is extremely large. However, we also believe that the use of current knowledge of psychology and neuroscience, as well as the use of heuristics such as the assumption of near-decomposability, helps cut down the size of this space. Moreover, evolutionary computation is one of the best methods to efficiently tackle large search spaces. Finally, new technologies such as grid computing may help mitigate the time needed to carry out search (Gobet & Lane, 2005).

*2. The amount of information hampers the search process.* We suggest that, on the contrary, the amount of information available constrains the search process, and then helps weed out poor candidate theories (cf. Newell, 1990).

*3. Humans can do it without the help of this or of similar methodology.* As noted in the introduction, information overload is a serious concern for psychology and neuroscience, and, indeed for most sciences. In addition, humans, including scientists, are prone to cognitive limitations that may seriously hamper

progress (Langley et al., 1987; Richman, Gobet, Staszewski, & Simon, 1996; Simon, 1989).

*4. Too much bias is introduced by the selection of primitives, tasks, and fitness function.* It is true that the choice of primitives for the brain structures and cognitive operations, as well as the choice of experiments and fitness function, determines and therefore biases the space of possible theories. It is also true that we may not have the right primitives and thus are searching in the wrong solution subspace. However, the same applies for psychologists and neuroscientists attempting to build these mappings, who, as we have just seen in the previous objection, may be ill-equipped computationally for carrying out this type of task.

*5. Setting up the databases used for evolving theories represents a huge and tedious task.* We agree with this point, but note that such databases would be useful on their own. Indeed, a substantial part of neuroinformatics currently consists in setting up databases encoding information about neuro-anatomical structures, neurophysiological processes, and behavioural results, to make that information available to other researchers. Even if the specific methodology proposed in this paper is found wanting, the databases can be reused in other projects aimed at modelling these data.

*6. The methodology is purely inductive and unlikely to be successful.* The first statement is not correct. As this methodology uses the current state of

scientific knowledge in the field, it is not “theory free,” but anchored in current neuroscientific theories.

### **Conclusion**

A major advantage of the methodology presented in this paper is that it enables selecting evolved theories that are consistent with most results contained in a given database. This avoids the selection biases that are unavoidable in informal theorizing and suggests an increased likelihood of finding theories to account for the empirical data while satisfying the constraints imposed. Another advantage is that the developed theories meet the criterion of sufficiency: being implemented as computer programs, they can indeed carry out the tasks under study.

To our knowledge, this constitutes the first attempt to use artificial evolution to tackle the problem of structure-to-function mapping in neuroscience. While we do not deny that a number of conceptual and practical questions remain to be answered, we believe that the potential is high for this methodology. First, the techniques described here, illustrated by the study of the prefrontal cortex, can be easily generalized to other regions of the brain. Second, these techniques can also be applied to domains where the structure-to-function mapping relates to a finer level of analysis (e.g., neuronal and molecular level); indeed, the exact level of explanation offered by these theories will depend on the primitives selected.

Third, while the focus in this paper is at the functional level, it is possible to further implement functions using neural nets. Fourth, the techniques can be used for each subcomponent of mapping theories in isolation (i.e., either functions or structures). Fifth, they can be used in other sciences as well, as long as theories can be formalized as computer programs and matched to a database of empirical results.

Philosophers of science have often noted the evolutionary character of scientific knowledge (Toulmin, 1967; Popper, 1979). In this paper, we have proposed to use the evolutionary process not only as a description of the way science evolves, but also as a tool that can actively evolve new scientific theories.

## References

- Angeline, P. J., & Pollack, J. B. (1992). The evolutionary induction of subroutines, *Proceedings of the fourteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.
- Bollobas, B., & Riordan, O. (1998). On some conjectures of Graffiti. *Discrete Mathematics*, 179, 223-230.

- Burgess, P. W., & Shallice, T. (1996). Response suppression, initiation and strategy use following frontal lobe lesions. *Neuropsychologia*, 34, 263-273.
- Chao, L. L., Haxby, J. V., & Martin, A. (1999). Attribute-based neural substrates in temporal cortex for perceiving and knowing about objects. *Nature Neuroscience*, 2, 913-919.
- Churchland, P. S., & Sejnowski, T. J. (1992). *The computational brain*. Cambridge, MA: MIT Press.
- Cohen, J. D., Braver, T. S., & O'Reilly, R. C. (1998). A computational approach to prefrontal cortex, cognitive control, and schizophrenia: Recent developments and current challenges. In A. C. Roberts & T. W. Robbins & L. Weiskrantz (Eds.), *The prefrontal cortex. Executive and cognitive functions* (pp. 195-220). Oxford: Oxford University Press.
- Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24, 87-114.
- Dehaene, S., & Changeux, J.-P. (1995). Neuronal models of prefrontal cortical function. *Ann. N. Acad. Sci.*, 769, 305-319.
- Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. (1996). *Rethinking innateness. A connectionist perspective on development*. Cambridge, MA: MIT Press.



- Gholson, B., Shadish, W., Neimeyer, R.A. & Houts, A. (1989). *Psychology of science: Contributions to metascience*. Cambridge, UK: Cambridge University Press.
- Gobet, F. (2001). Is experts' knowledge modular? *Proceedings of the 23rd Meeting of the Cognitive Science Society* (pp. 336-431). Mahwah, NJ: Erlbaum.
- Gobet, F., & Clarkson, G. (2004). Chunks in expert memory: Evidence for the magical number four... or is it two? *Memory*, 12, 732-747.
- Gobet, F., & Lane, P. C. R. (2005). A distributed framework for semi-automatically developing architectures of brain and mind. *Proceedings of the First International Conference on e-Social Science*. Manchester: ESRC National Centre for e-Social Science.
- Gobet, F., & Simon, H. A. (2000). Five seconds or sixty? Presentation time in expert memory. *Cognitive Science*, 24, 651-682.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.
- Hadamard, J. (1945). *The psychology of invention in the mathematical field*. New York: Dover.
- Hanson, N.R. (1958). *Patterns of discovery*. Cambridge, UK: Cambridge University Press.

- Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Cambridge, MA: MIT Press.
- Kanehisa, M. (2000). *Post-genome informatics*. Oxford: Oxford University Press.
- Kentala, E., Laurikkala, J., Pyykko, I., & Juhola, M. (1999). Discovering diagnostic rules from a neurologic database with genetic algorithms. *Annals of Otology, Rhinology and Laryngology*, 108, 948-954.
- Koslow, S. H., & Huerta, M. F. (1997). *Neuroinformatics*. Mahwah, NJ: Erlbaum.
- Kosslyn, S. M., & Koenig, O. (1992). *Wet mind*. New York: The Free Press.
- Koza, J. R. (1992). *Genetic programming. On the programming of computers by means of natural selection*. Cambridge, MA: MIT Press.
- Koza, J. R. (1994). *Genetic programming II. Automatic discovery of reusable programs*. Cambridge, MA: MIT Press.
- Kuhn, T. S. (1977). *The essential tension. Selected studies in scientific tradition and change*. Chicago: The University of Chicago Press.
- Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). *Scientific discovery*. Cambridge, MA: MIT press.
- Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.

- Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
- Parker, A., & Gaffan, D. (1998). Interaction of frontal and perirhinal cortices in visual object recognition memory in monkeys. *European Journal of Neuroscience*, 10, 3044-3057.
- Pattichis, C. S., & Schizas, C. N. (1996). Genetics-based machine learning for the assessment of certain neuromuscular disorders. *IEEE Transactions on Neural Networks*, 7, 427-439.
- Popper, K. R. (1979). *Objective knowledge. An evolutionary approach*. Oxford: Clarendon Press.
- Purpura, D. (1997). The human brain project: Priority or problem? In S. H. Koslow & M. F. Huerta (Eds.), *Neuroinformatics*. Mahwah, NJ: Erlbaum.
- Richman, H. B., Gobet, F., Staszewski, J. J., & Simon, H. A. (1996). Perceptual and memory processes in the acquisition of expert performance: The EPAM model. In K. A. Ericsson (Ed.), *The road to excellence* (pp. 167-187). Mahwah: Erlbaum.
- Roberts, A. C., Robbins, T. W., & Weiskrantz, L. (Eds.). (1998). *The prefrontal cortex. Executive and cognitive functions*. Oxford: Oxford University Press.
- Shallice, T. (1990). *From neuropsychology to mental structure*. Cambridge: Cambridge University Press.

- Simon, H. A. (1977). *Models of discovery and other topics in the methods of science*. Dordrecht: Reidel.
- Simon, H. A. (1989). *Models of thought (vol II)*. New Haven: Yale University Press.
- Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge: MIT Press.
- Simonton, D. K. (1999). *Origins of genius*. Oxford: Oxford University Press.
- Sonka, M., Tadikonda, S. K., & Collins, S. M. (1996). Knowledge-based interpretation of MR brain images. *IEEE Transactions on Medical Imaging*, 15, 443-452.
- Toulmin, S. (1967). The evolutionary development in natural science. *American Scientist*, 55, 456-471.
- Uttal, W. R. (2001). *The new phrenology: The limits of localizing cognitive processes in the brain*. Cambridge, MA: The MIT Press.
- Valdés-Pérez, R. E. (1999). Principles of human-computer collaboration for knowledge discovery in science. *Artificial Intelligence*, 107, 335-346.
- Xu, Y., Mural, R., Einstein, J., Shah, M., & Eberbacher, E. (1996). GRAIL, A multi-agent neural network system for gene identification. *Proceedings of the IEEE*, 84, 1544-1552.

### **Authors' Note**

This research was supported by a grant from Brunel University. We thank Enrique Frias-Martinez for help in implementing the software used for evolving theories.

Table 1.

*Mean and standard deviation of percentages of the “animal” and “tool” conditions, for the human data (Chao et al., 1999) and one of the evolved theories.*

		Measures			
		Observed	Evolved	Deviation	Hit
Condition			theory	percentage	
“Animal”					
	mean	97	92	5.1	yes
	sd	1.4	2.1	50.0	no
“Tool”					
	mean	95	93	2.1	yes
	sd	1.2	3.1	158.3	no

### Figure Captions

**Figure 1.** Examples of genetic-programming trees.

**Figure 2.** Crossover in genetic programming.

**Figure 3.** Some properties of *mapping theories*. A given structure may map to several functions (narrow dashed arrows), conversely, a given function may map to several structures (thick dashed arrows). A group of nodes denoting structures may be considered as a unit (subtree) mapping to a single function, and vice-versa (ellipses).

**Figure 4.** Example of an evolved functional theory. Top part: Lisp code of the evolved theory; bottom part: diagrammatic representation of this theory. Three primitive are used. `PutStm` puts a new element into STM. `Progn2` allows the execution of two subtrees one after the other. `Compare23` compares element 2 and element 3 of STM.

**Figure 5.** An illustration of how the functional theory of Figure 4 could be linked to brain structures. PFC = prefrontal cortex.

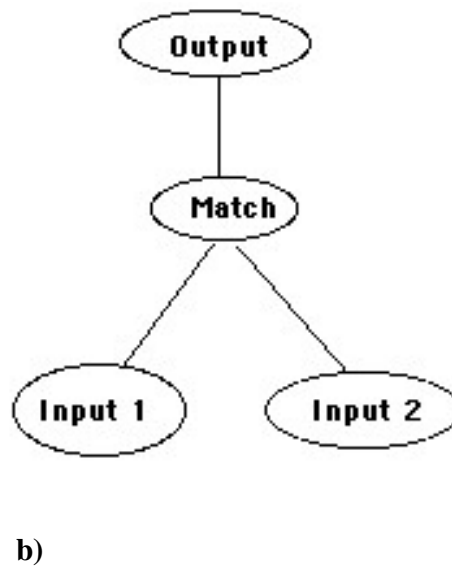
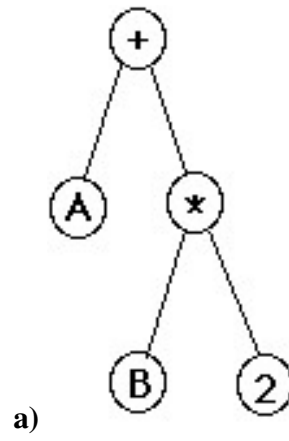


Figure 1



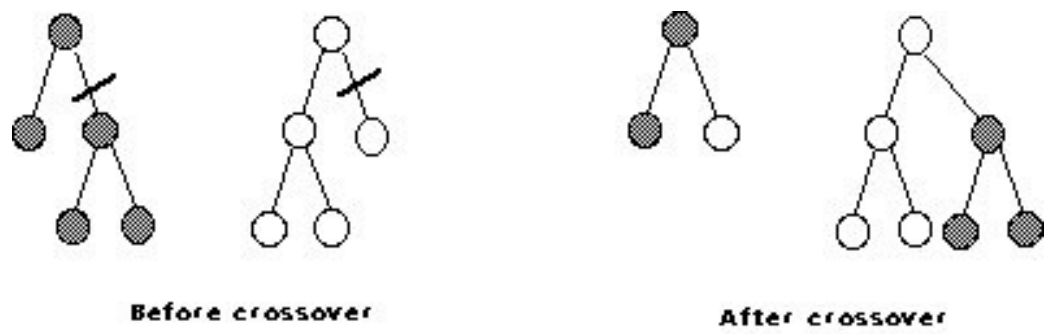


Figure 2.

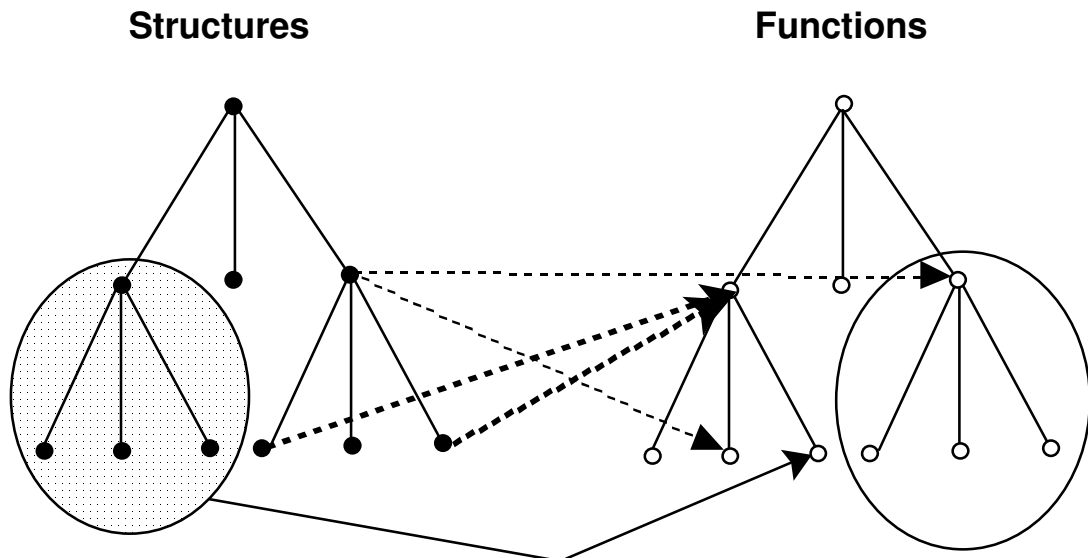


Figure 3.

```
(Progn2
  (Progn2
    (PutStm Input1)
    (Progn2 (PutStm Input2)
      (PutStm Input3)))
  (Compare23))
```

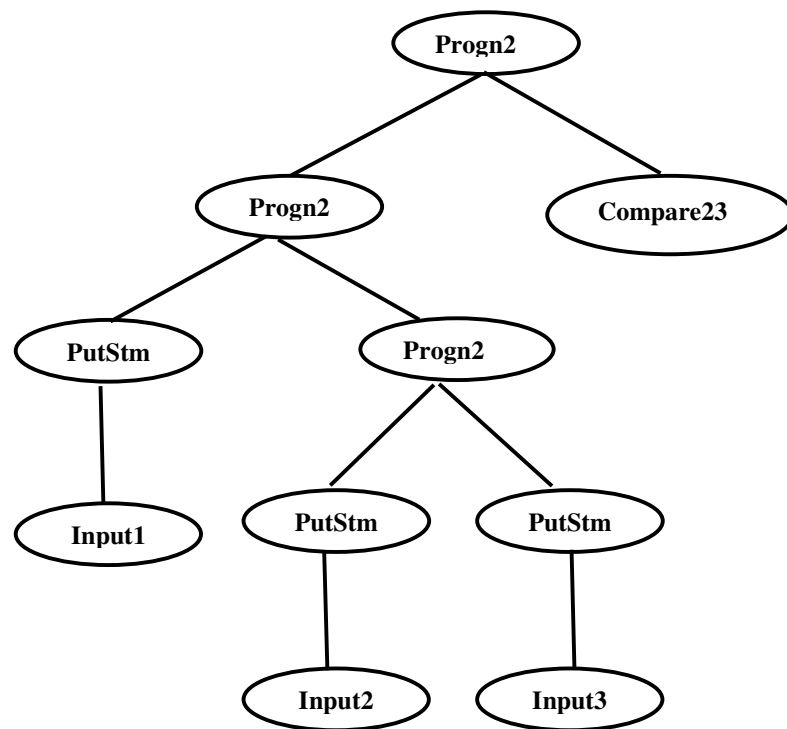
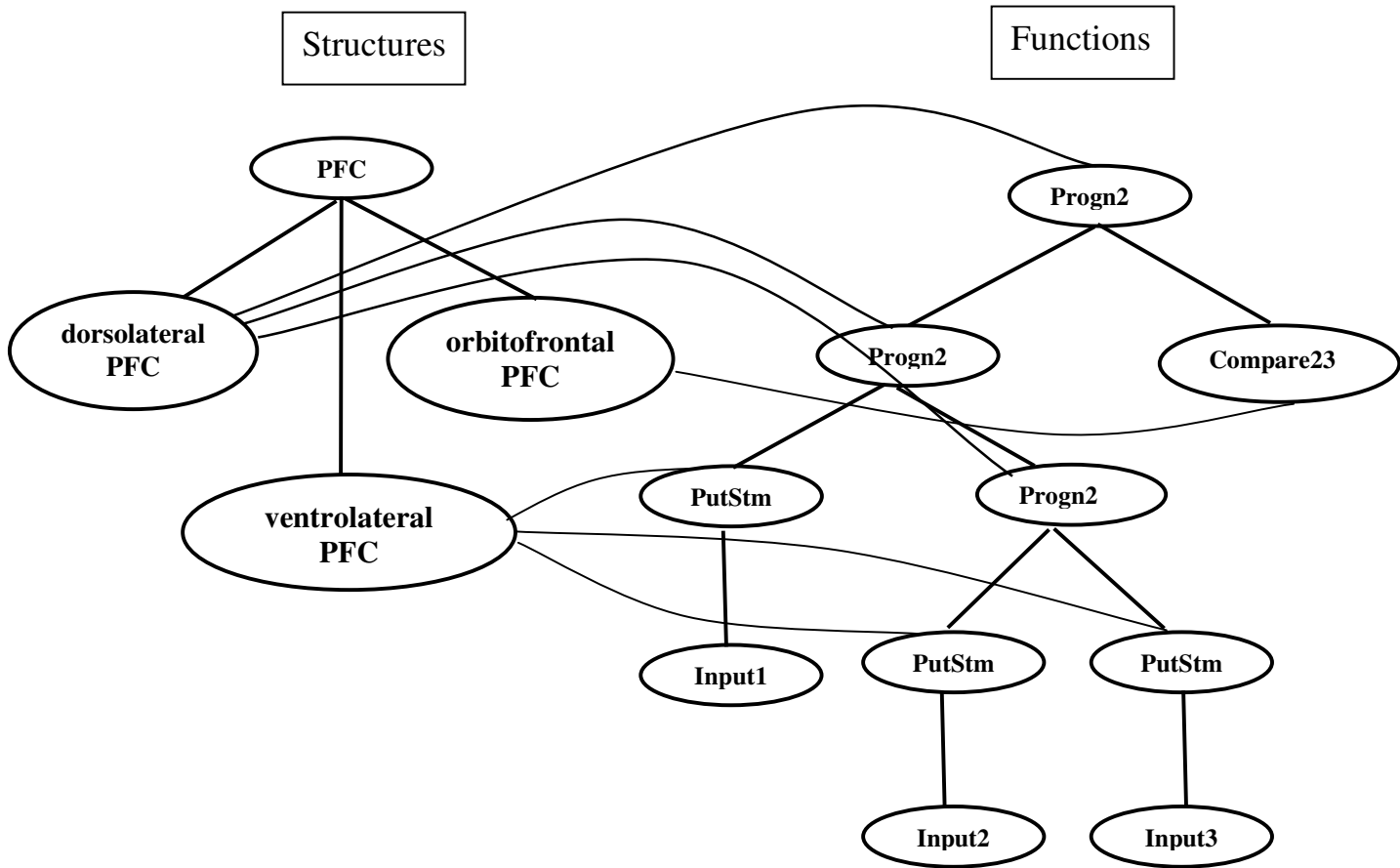


Figure 4

**Figure 5**